

Aula 07

Estruturas de dados

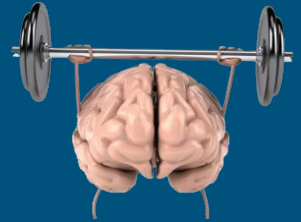
emerson@paduan.pro.br

Antes de
iniciar...

Dúvidas ???



Exercício 6-6



1. Criar uma classe `Conta`, que possua um número, um saldo e os métodos para obter dados da conta, depositar e sacar.
2. Crie as subclasses da classe `Conta`: `ContaCorrente`, `ContaEspecial` e `ContaPoupanca`.
A `ContaCorrente` permite fazer saques somente se houver saldo suficiente. A `ContaEspecial` possui um limite que permite fazer saques se o saldo mais o limite da conta cobrir o valor pretendido de saque. A `ContaPoupanca` faz saque se houver saldo, mas há uma taxa por operação. Além disso, a `ContaCorrente` deve reescrever o método `deposita`, com o objetivo de retirar uma taxa bancária de dez centavos de cada depósito.
3. Crie a classe `GerenciaConta` que permite adicionar, remover, listar, e fazer os saques e depósitos.
4. Crie uma classe `AppContas` com o método `main` contendo um menu com opções para realizar operações nas contas a partir da classe `GerenciaContas`.

emerson@paduan.pro.br

Classe abstrata

As classes abstratas não permitem instanciar objetos. São classes feitas especialmente para serem modelos para suas classes derivadas.

Métodos abstratos presentes na classe abstrata, obriga a classe filha a definir tais métodos, pois, caso contrário, a classe filha também se tornará abstrata.

emerson@paduan.pro.br

Static

Aplicado a atributos: um único atributo para todas as instâncias (objetos) da classe

Aplicado a métodos: Pode-se utilizar os métodos sem que seja necessário instanciar um objeto.

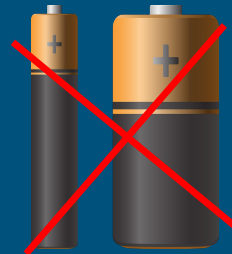
emerson@paduan.pro.br

Estruturas de Dados



emerson@paduan.pro.br

Pilha



Não é a bateria

emerson@paduan.pro.br

O que é?



school916.ucoz.net



emerson@paduan.pro.br

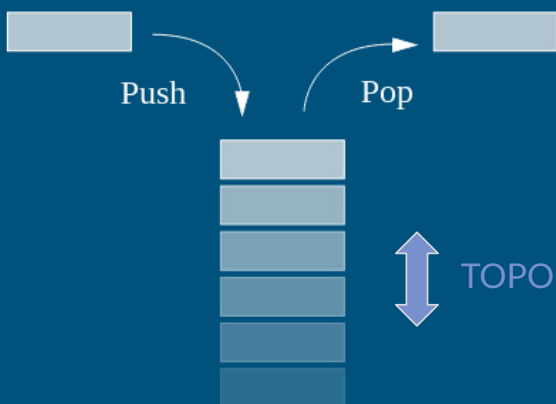
Pilha

- ✓ Conjunto ordenado de itens no qual somente em uma das extremidades novos itens podem ser inseridos, ou itens podem ser removidos.
- ✓ A extremidade onde os itens são inseridos ou removidos chama-se **Topo** da pilha.
- ✓ **FILO** – First In, Last Out : O primeiro a entrar será o último a sair.
- ✓ **LIFO** – Last In, First Out : O último a entrar será o primeiro a sair.



emerson@paduan.pro.br

Operações



- ✓ Pilha vazia → **isEmpty()**
- ✓ Inserir (empilhar) → **push()**
- ✓ Remover (desempilhar) → **pop()**
- ✓ Informa o elemento no topo → **peek()**
- ✓ Quantidade de elementos → **size()**

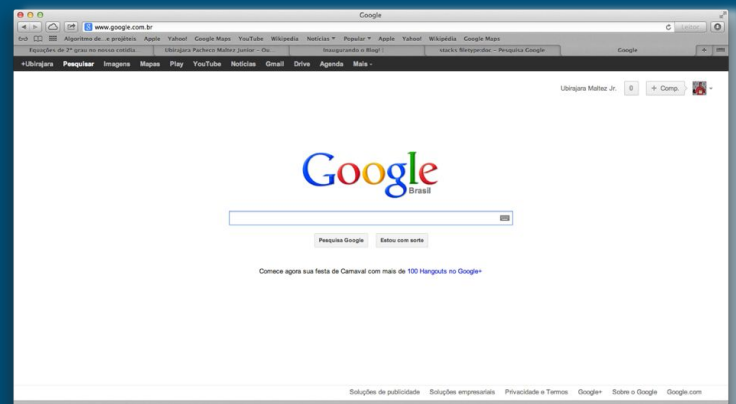
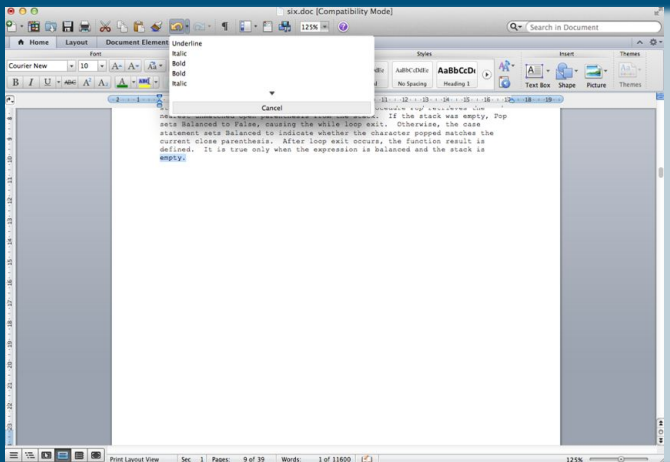
emerson@paduan.pro.br

Aplicações



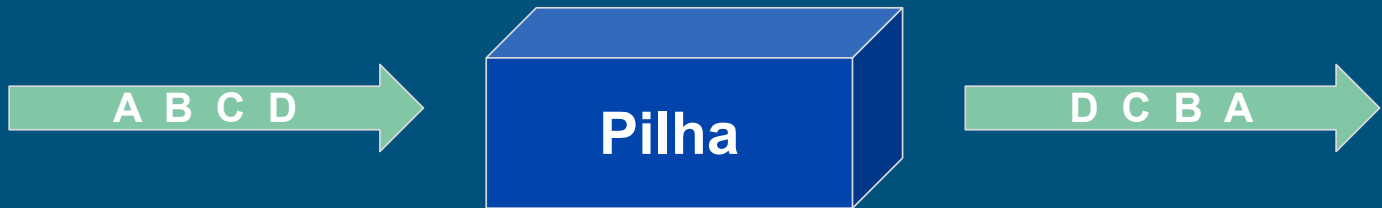
emerson@paduan.pro.br

Aplicações



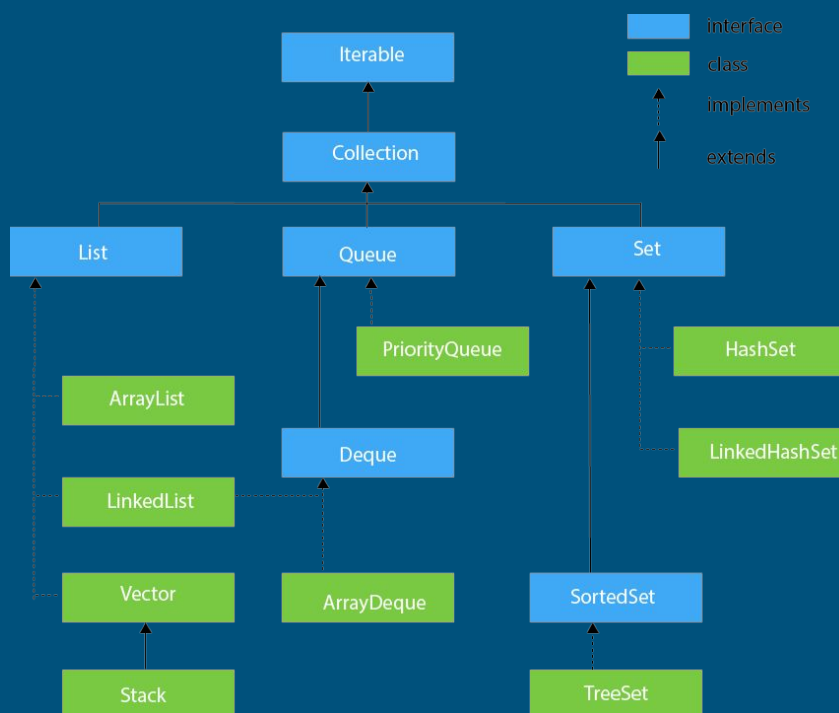
emerson@paduan.pro.br

Utilizando uma pilha



emerson@paduan.pro.br

Classes para ED em Java



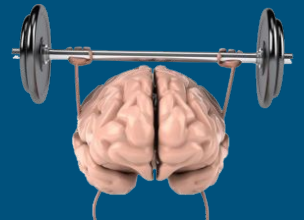
emerson@paduan.pro.br

Let's Code!

```
$this->rule_exists( $resource_details['id'], $role_details['id'] ) {  
    if ( $access == false ) {  
        // Remove the rule as there is currently no need for it  
        $details['access'] = !$access;  
        $this->_sql->delete( 'acl_rules', $details );  
    } else {  
        // Update the rule with the new access value  
        $this->_sql->update( 'acl_rules', array( 'access' => $access ) );  
    }  
    foreach( $this->rules as $key=>$rule ) {  
        if ( $details['role_id'] == $rule['role_id'] && $details['resource_id'] == $rule['resource_id'] ) {  
            if ( $access == false ) {  
                unset( $this->rules[ $key ] );  
            } else {  
                $this->rules[ $key ]['access'] = $access;  
            }  
        }  
    }  
}
```

emerson@paduan.pro.br

Exercício 7-1



Escreva um programa (main) que utilize uma Pilha para inverter uma frase escrita pelo usuário.

Por exemplo, para a String:

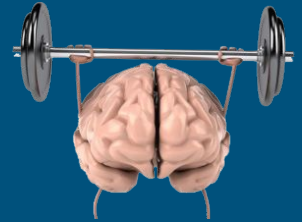
“ESTE EXERCICIO E MUITO FACIL”

o resultado deve ser:

“LICAF OTIUM E OICICREXE ETSE.

emerson@paduan.pro.br

Exercício 7-2



Escreva um programa (main) que utilize uma Pilha para inverter uma frase escrita pelo usuário, preservando a ordem das palavras.

Por exemplo, para a String:

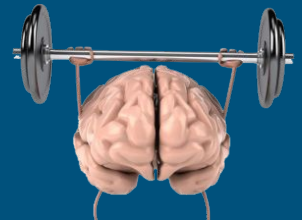
“ESTE EXERCICIO E MUITO FACIL”

o resultado deve ser:

“ETSE OICICREXE E OTIUM LICAF.

emerson@paduan.pro.br

Exercício 7-3



Digamos que nosso alfabeto contém apenas as letras a, b e c.

Considere o seguinte conjunto de Strings sobre nosso alfabeto:

c, aca, bcb, abcba, bacab, aacaa, bbcbb...

Qualquer String desse conjunto tem a forma WcM, sendo W uma sequência de letras que só contém a e b, e M o inverso de W (ou seja, M é W lido de trás para a frente).

Escreva um programa em Java que decida se uma String X pertence ou não ao nosso conjunto, ou seja, decida se X é da forma WcM.

emerson@paduan.pro.br

Fila



Fila de espera

emerson@paduan.pro.br

Fila

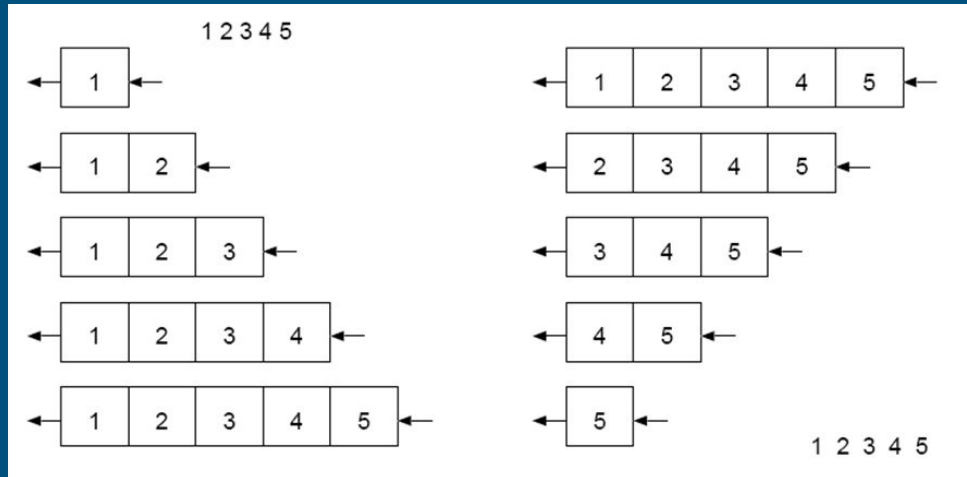
Conjunto ordenado de itens no qual somente em UMA das extremidades novos itens podem ser inseridos, E somente na OUTRA extremidade itens podem ser removidos.

FIFO – First In, First Out - O primeiro a entrar será o primeiro a sair

A extremidade onde os itens são removidos é o começo (início) da fila e a extremidade onde são inseridos é o final da fila.

emerson@paduan.pro.br

Comportamento



emerson@paduan.pro.br

Let's Code!

```
$this->rule_exists( $resource_details['id'], $role_details['id'] ) {
    if ( $access == false ) {
        // Remove the rule as there is currently no need for it
        $details['access'] = $access;
        $this->sql->delete( 'acl_rules', $details );
    } else {
        // Update the rule with the new access value
        $this->sql->update( 'acl_rules', array( 'access' => $access ) );
    }
}

foreach( $this->rules as $key => $rule ) {
    if ( $details['role_id'] == $rule['role_id'] && $details['resource_id'] == $rule['resource_id'] ) {
        if ( $access == false ) {
            unset( $this->rules[ $key ] );
        } else {
            $this->rules[ $key ]['access'] = $access;
        }
    }
}
```

emerson@paduan.pro.br

Exemplo

```
public static void main(String[] args) {
    Queue<Pessoa> queueA = new LinkedList<>();

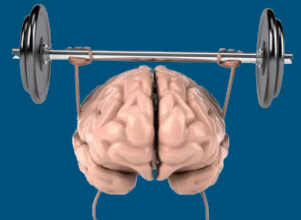
    Random rd = new Random();

    for (int i = 0; i < 5; i++) {
        queueA.add(new Pessoa(rd.nextInt(10), "Pessoa "+ i ));
    }

    System.out.println(queueA);
}
```

emerson@paduan.pro.br

Exercício 7-3



Crie uma classe FilaDePrioridade para armazenar objetos da classe Pessoa (que contém nome e senha número de atendimento), simulando a chegada e saída de pessoas em uma fila de banco. A classe deve utilizar internamente 2 filas (uma normal e uma de prioridade) utilizando a classe LinkedList (Queue) apresentada nesta aula como base.

Ao realizar a remoção da fila, deve ser seguida a seguinte estratégia: devem ser atendidas 2 com prioridade para cada 1 pessoal normal. Caso a fila de prioridades esteja vazia, remova pessoas normais.

Para testar, construa um programa que utilize essa classe FilaDePrioridade, fornecendo opções para a entrada de pessoas do tipo normal e prioritária, atender o próximo da fila, e exibir a fila.

emerson@paduan.pro.br

Árvores



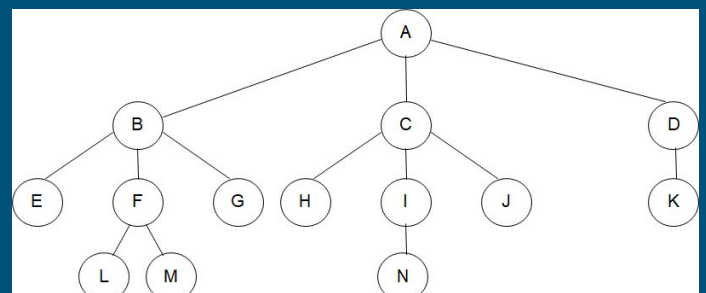
E as árvores ...

emerson@paduan.pro.br

Árvores

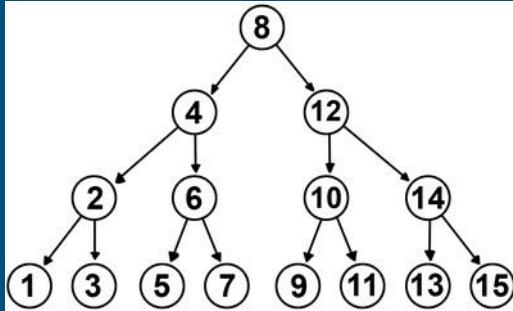
Árvores são estruturas de dados formadas por um conjunto finito de elementos denominados nós.

Temos um nó especial chamado raiz da árvore, e os elementos restantes são particionados em conjuntos distintos não vazios, as subárvores, sendo cada um destes conjuntos por sua vez uma árvore.

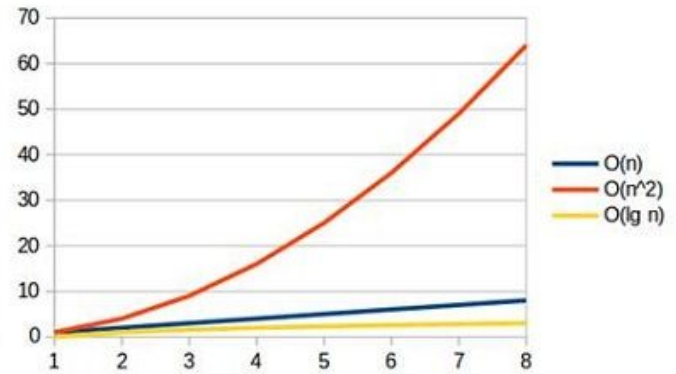


emerson@paduan.pro.br

Vantagem



| $O(n)$ | $O(n^2)$ | $O(\lg n)$ |
|--------|----------|------------|
| 1 | 1 | 0 |
| 2 | 4 | 1 |
| 3 | 9 | 1.6 |
| 4 | 16 | 2 |
| 5 | 25 | 2.3 |
| 6 | 36 | 2.6 |
| 7 | 49 | 2.8 |
| 8 | 64 | 3 |



emerson@paduan.pro.br

Let's Code!

```
$this->rule_exists( $resource_details['id'], $role_details['id'] ) {
    if ( $access == false ) {
        // Remove the rule as there is currently no need for it
        $details['access'] = false;
        $this->sql->delete( 'acl_rules', $details );
    } else {
        // Update the rule with the new access value
        $this->sql->update( 'acl_rules', array( 'access' => $access ) );
    }
}

foreach( $this->rules as $key=>$rule ) {
    if ( $details['role_id'] == $rule['role_id'] && $details['resource_id'] == $rule['resource_id'] ) {
        if ( $access == false ) {
            unset( $this->rules[ $key ] );
        } else {
            $this->rules[ $key ]['access'] = $access;
        }
    }
}
```

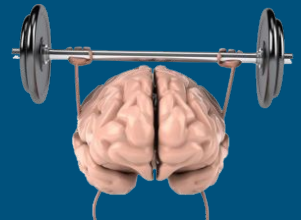
emerson@paduan.pro.br

Em Java

```
public static void main(String[] args) {  
    int[] nums={2,4,1,6,3,7,9,5};  
    TreeSet<Integer> tree=new TreeSet<>();  
    //Insere todos os elementos na árvores  
    for (int num : nums) {  
        tree.add(num);  
    }  
  
    //Exibe todos os elementos da árvores  
    for(int s: tree){  
        System.out.println(s);  
    }  
}
```

emerson@paduan.pro.br

Exercício 7-4



Suponhamos que tenhamos uma lista com 1000 números aleatórios entre 1 e 500, e precisamos remover os números duplicados. Uma maneira é comparar cada número da lista com todos os demais. Isto seria um algoritmo com alto índice de comparações e que demoraria para ser executado. Outra possibilidade é utilizar uma árvore, e inserir os números da lista. Como a árvore não permite repetições, somente os números diferentes seriam inseridos. Ao final bastaria exibir os valores da árvore.

Escreva um programa para esta simulação.

emerson@paduan.pro.br